計算の理論 II
帰納的関数

月曜4校時
大月美佳

講義の前に

- 前回のミニテスト解答
  (**2の概略のみ**)

今日の講義内容

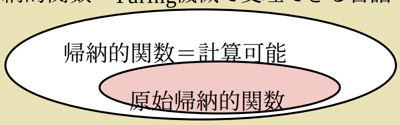
1. 原始帰納的関数
   1. 初期関数
   2. 合成と原始帰納
2. 原始帰納的でない関数
   1. Ackermann関数
3. 原始帰納的な集合と述語
4. 帰納的関数と部分帰納的関数

原始帰納的関数

- 計算可能な関数の一部
- 原始帰納的関数

  ー拡大→帰納的関数＝計算できる関数の族

  帰納的関数＝Turing機械で受理できる言語



数論的関数

- 自然数（非負整数）N

  N={0, 1, 2, 3, …}
- 数論的関数(関数)

  N個の自然数の組に対して、

  高々1個の自然数を対応づける関数

  $f: N^n \rightarrow N$

初期関数

- 初期関数に操作を加えて原始帰納的関数を作成する。

(1) $Z(x)=0$

どんなxも0にしてしまう。

(2) $S(x)=x+1$

xに1を加える。

(3) $U_n^i(x_1, \ldots, x_i, \ldots, x_n)=x_i$

i番目のxiを取り出す。

合成と原始帰納

- 初期関数に加える操作

(I) 合成

r変数の関数hとr個のn変数関数g_i(1≦i≦r)から、
n変数の関数fを以下の操作で作ること。
$f(x_1, \ldots, x_n)=h(g_1(x_1, \ldots, x_n), \ldots, g_r(x_1, \ldots, x_n))$

(II) 原始帰納 (primitive recursion)

n-1変数の関数gとn+1変数の関数hから、
n変数の関数fを以下の操作で作ること。
$f(x_1, \ldots, x_n)=g(x_1, \ldots, x_{n-1})$ (x_n=0のとき)
$f(x_1, \ldots, x_n)=h(x_1, \ldots, x_{n-1}, x_n, f(x_1, \ldots, x_n-1))$ (x_n＞0のとき)

原始帰納的関数 (primitive recursive)

- 定義

初期関数(1), (2), (3)に
操作(I), (II)を
有限回(0回以上)適用して
得られた関数。

原始帰納的関数の例 (4)

(4) 定数関数 $C_n^k(x_1, \ldots, x_i, \ldots, x_n)=k$

なぜならば

$C_n^k(x_1, \ldots, x_i, \ldots, x_n)$
$=S(S(\ldots S(Z(U_n^i(x_1, \ldots, x_n)))\ldots))=k$

0にk回1を加算

xを1個取り出す
(どれでも良い)

選ばれたxを0にする

原始帰納的関数の例 (5)

(5) x_1+x_2

$plus(x_1, x_2)= x_1+x_2$とおくと、

$plus(x_1, x_2)= U_1^1(x_1)$ g(x1)
(x_2=0のとき) h(x1, x2, f(x1, x2-1))
$plus(x_1, x_2)=S(U_3^3(x_1, x_2, plus(x_1, x_2-1)))$
(x_2＞0のとき)

plus(x1, x2)の結果を取り出す
plus(x1, x2)に1を加算

原始帰納的関数の例 (6)

(6) $x_1 \cdot x_2$

$times(x_1, x_2)= x_1 \cdot x_2$とおくと、

$times(x_1, x_2)=Z(x_1)=0$ g(x1) (x_0=0のとき)
$times(x_1, x_2)=p(x_1, x_2, times(x_1, x_2-1))$ (x_2＞0のとき)

ここで、 h(x1, x2, f(x1, x2-1))

$p(x, y, z)= plus(U_3^1(x, y, z), U_3^3(x, y, z))$
$=x+z$

原始帰納的関数の例 (7)

(7) x^y

$power(x_1, x_2)= x_1^{x_2}$とおくと、 g(x1)

$power(x_1, x_2)=S(Z(x_1))=1$ (x_2=0のとき)
$power(x_1, x_2)=p(x_1, x_2, power(x_1, x_2-1))$
$= x_1 \cdot power(x_1, x_2-1)$ (x_2＞0のとき)

ここで、 h(x1, x2, f(x1, x2-1))

$p(x, y, z)= times(U_3^1(x, y, z), U_3^3(x, y, z))$
$=x \cdot z$

原始帰納的関数の例 (8)

(8) $x_1!$

factorial(x_1)= $x_1!$ とおくと、 g(x_1)

factorial(x_1)=S(Z(x_1))=1　　(x_1=0のとき)

factorial(x_1)=p(x_1, factorial(x_1-1))

= x_1 ･factorial(x_1-1) (x_1＞0のとき)

ここで、　h(x_1, f(x_1-1))

p(x, y)= times(U_2^1(x, y)), U_2^2(x, y))

=x ･y

原始帰納的関数の例 (9)

(9) pd(x_1)を

pd(x_1)=0　　(x_1=0のとき)

pd(x_1)= x_1-1　　(x_1＞0のとき)

とおくと、 g(x_1)

pd(x_1)=Z(x_1)　(x_1=0のとき)

pd(x_1)=p(x_1, pd(x_1-1)) (x_1＞0のとき)

ここで、　h(x_1, f(x_1-1))

p(x, y)= U_2^2(x, y)=y

原始帰納的関数の例 (10)

(10) 自然数上での減算 $x_1 \dot{-} x_2$ を

$x_1 \dot{-} x_2 = x_1 - x_2$ (x_1≧x_2のとき)

$x_1 \dot{-} x_2 = 0$　　(x_1＜x_2のとき)

とする。 $x_1 \dot{-} x_2$ =n-minus(x_1, x_2) とおくと、

n-minus(x_1, x_2)= U_1^1(x_1) (x_2=0のとき)

n-minus(x_1, x_2)=p(x_1, x_2, n-minus(x_1, x_2-1))

g(x_1)　=pd(n-minus(x_1, x_2 -1)) (x_2＞0のとき)

ここで、　h(x_1, x_2, f(x_1, x_2-1))

p(x, y, z)= pd(U_3^3(x, y, z))=pd(z)

原始帰納的関数の例 (11)

(11) 差の絶対値| x_1-x_2 |を

| x_1-x_2 | = x_1-x_2　　(x_1≧x_2のとき)

| x_1-x_2 | = x_2-x_1　　(x_1＜x_2のとき)

とする。| x_1-x_2 |=abs-minus(x_1, x_2)とおくと、

abs-minus(x_1, x_2)= U_1^1(x_1) (x_2=0のとき)

abs-minus(x_1, x_2)=p(x_1, x_2, abs-minus(x_1, x_2-1)) (x_2＞0のとき)

ここで、　g(x_1)　h(x_1, x_2, f(x_1, x_2-1))

p(x, y, z)= n-minus(U_3^1(x, y, z), U_3^2(x, y, z))

+ n-minus(U_3^2(x, y, z), U_3^1(x, y, z))

=n-minus(x, y)+ n-minus(y, x)

原始帰納的関数の例 (12)

(12) xの符号を表す関数

sd(x_1)=0　(x_1=0のとき)

sd(x_1)=1　(x_1＞0のとき)

とすると、

sd(x_1)=Z(x_1)　　(x_1=0のとき)

sd(x_1)=S(Z(U_2^1(x_1, sd(x_1-1))) (x_1＞0のとき)

原始帰納的関数その他 1

関数f(x_1, …, x_n, y)が原始帰納的であれば、有限和も有限積も原始帰納的である。

$\underset{y?0}{?}\ f(x_1,\ldots, x_n, y)\ ?\ 0$

$\underset{y?z}{?}\ f(x_1,\ldots, x_n, y)\ ?\ f(x_1,\ldots, x_n, 0)\ ?\ \ldots\ ?\ f(x_1,\ldots, x_n, z\ ?\ 1)$

$\underset{y?0}{?}\ f(x_1,\ldots, x_n, y)\ ?\ 0$

$\underset{y?z}{?}\ f(x_1,\ldots, x_n, y)\ ?\ f(x_1,\ldots, x_n, 0)\ ?\ \ldots\ ?\ f(x_1,\ldots, x_n, z\ ?\ 1)$

原始帰納的関数その他 2

関数 $f(x_1, \ldots, x_n, y)$ が原始帰納的であれば、以下も原始帰納的である。

$$\mathop{?}_{y?z} f(x_1,\ldots,x_n,y),\ \mathop{?}_{y?z} f(x_1,\ldots,x_n,y),$$

$$\mathop{?}_{u?y?z} f(x_1,\ldots,x_n,y),\ \mathop{?}_{u?y?z} f(x_1,\ldots,x_n,y),$$

$$\mathop{?}_{u?y?z} f(x_1,\ldots,x_n,y),\ \mathop{?}_{u?y?z} f(x_1,\ldots,x_n,y),$$

原始帰納的でない関数

計算可能だが原始帰納的ではない関数

F(x, n)

任意の1変数の原始帰納的関数f(x)に対して

f(x)=F(n, x)

となる自然数が存在するような関数。

→証明

Ackerman関数

原始帰納的でない関数

A(x, y)

1. A(0, y) = y+1
2. A(x+1, 0) = A(x, 1)
3. A(x+1, y+1) = A(x, A(x+1, y))

計算に手間のかかる関数

→例 A(2, 2)

原始帰納的な集合と述語

特徴関数 C_S, C_P

集合 $S \in N^n$

$C_S(x_1, \ldots, x_n)=0$ $((x_1, \ldots, x_n) \in S$ のとき)

$C_S(x_1, \ldots, x_n)=1$ $((x_1, \ldots, x_n) \in S$ のとき)

が原始帰納的であるとき、Sは原始帰納的集合。

述語 $P(x_1, \ldots, x_n)$

$C_P(x_1, \ldots, x_n)=0$ $(P(x_1, \ldots, x_n)$ のとき)

$C_P(x_1, \ldots, x_n)=1$ $(\neg P(x_1, \ldots, x_n)$ のとき)

が原始帰納的であるとき、Pは原始帰納的述語。

原始帰納的集合の性質

集合 $S, R \in N^n$ が原始帰納的であれば、

$S=N^n-R$

$S \cup R$

$S \cap R$

も原始帰納的。

→証明

原始帰納的述語の性質 1

$P(x_1, \ldots, x_r)$ を原始帰納的述語とし、

$h_1, \ldots, h_r$ をn変数の原始帰納的関数とする。

このとき述語

$P(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n))$

は原始帰納的。

→証明

原始帰納的述語の性質 2

$g_1, \ldots, g_{m+1}$をn変数の原始帰納的関数とし、
$P_1, \ldots, P_m$をn変数の原始帰納的述語で、
各$(x_1, \ldots, x_n)$に対して高々1個の$P_i(x_1, \ldots, x_n)$が真になるものとする。

このとき述語

$f(x_1, \ldots, x_n)=g_1(x_1, \ldots, x_n)$ $(P_1(x_1, \ldots, x_n)$のとき)

…

$f(x_1, \ldots, x_n)=g_m(x_1, \ldots, x_n)$ $(P_m(x_1, \ldots, x_n)$のとき)

$f(x_1, \ldots, x_n)=g_{m+1}(x_1, \ldots, x_n)$ (それ以外のとき)

は原始帰納的。→証明

原始帰納的述語の性質 3

$P(x_1, \ldots, x_n)$, $Q(x_1, \ldots, x_n)$を原始帰納的述語とすれば、述語

$\neg P(x_1, \ldots, x_n)$

$P(x_1, \ldots, x_n) \vee Q(x_1, \ldots, x_n)$

$P(x_1, \ldots, x_n) \wedge Q(x_1, \ldots, x_n)$

$P(x_1, \ldots, x_n) \rightarrow Q(x_1, \ldots, x_n)$

$P(x_1, \ldots, x_n) \rightarrow Q(x_1, \ldots, x_n)$

は原始帰納的。→証明

原始帰納的述語の性質 4

$P(x_1, \ldots, x_n, y)$を原始帰納的述語とすれば、述語

$(\exists y)_{<z}P(x_1, \ldots, x_n, y)$

$\Leftrightarrow P(x_1, \ldots, x_n, 0) \vee \ldots \vee P(x_1, \ldots, x_n, z-1)$

$(\forall y)_{<z}P(x_1, \ldots, x_n, y)$

$\Leftrightarrow P(x_1, \ldots, x_n, 0) \wedge \ldots \wedge P(x_1, \ldots, x_n, z-1)$

は原始帰納的。→証明

有界μ作用素

述語$P(x_1, \ldots, x_n, y)$に対して、
$(\mu y)_{<z}P(x_1, \ldots, x_n, y)$をn+1変数の以下のような関数として定義する。

$(\mu y)_{<z}P(x_1, \ldots, x_n, y)$

$=\min\{y|y<z \wedge P(x_1, \ldots, x_n, y)\}$

$((\exists y)_{<z}P(x_1, \ldots, x_n, y)$のとき)

$(\mu y)_{<z}P(x_1, \ldots, x_n, y)$

$=z$

$(\neg(\exists y)_{<z}P(x_1, \ldots, x_n, y)$のとき)

原始帰納的述語の性質 5

述語$P(x_1, \ldots, x_n, y)$が原始帰納的であれば$(\mu y)_{<z}P(x_1, \ldots, x_n, y)$は原始帰納的である。

→証明

原始帰納的述語と関数の例 1

(13) 述語$x=y$

(14) 述語$x<y$

(15) 述語$x \leqq y$

(16) 関数$\max(x, y)$

(17) 関数$\min(x, y)$

(18) 関数$\max(x_1, \ldots, x_n)$

(19) 述語$x|y$ (xはyを割り切る)

(20) 述語$P_r(x)$ (xは素数である)

原始帰納的述語と関数の例 2

(21) 述語x/y (xをyで割ったときの商)
(22) 第n+1番目の素数を表す関数p_n
(23) 関数
　l(a)=aの素因数分解における0でない数の個数
　　　(a≠0のとき)
　l(a)=0　(a=0のとき)
(24) aとiの関数
　$(a)_i$=aの素因数分解におけるp_iのべき数　(a≠0のとき)
　$(a)_i$=0　(a=0のとき)

原始帰納的述語と関数の例 3

(25) 関数

$$x\square y ? x ? \underset{i?l(y)}{?} p_{l(x)?i}^{(y)_i}$$

μ作用素
(μ-operator)

n+1変数の述語からn変数の関数を作る操作
定義
　述語$P(x_1, \ldots, x_n, y)$に対して
　$\mu yP(x_1, \ldots, x_n, y)$
　　$=\min\{y|P(x_1, \ldots, x_n, y)\}$
　　$((\exists y)P(x_1, \ldots, x_n, y)$のとき)
　$\mu yP(x_1, \ldots, x_n, y)$
　　=無定義
　　$(\neg(\exists y)P(x_1, \ldots, x_n, y)$のとき)

正則

述語$P(x_1, \ldots, x_n, y)$が正則
　＝ 任意の$(x_1, \ldots, x_n)$に対して$P(x_1, \ldots, x_n, y)$を真とするyが存在する。

関数$g(x_1, \ldots, x_n, y)$が正則
　＝任意の$(x_1, \ldots, x_n)$に対して$g(x_1, \ldots, x_n, y)=0$となるyが存在する。

2つの新操作

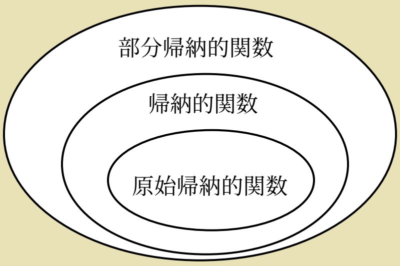
(III) 全域的であることが保証されない。
　関数$g(x_1, \ldots, x_n, y)$から関数$f(x_1, \ldots, x_n)$を以下の操作で作る。
　　$f(x_1, \ldots, x_n)=\mu y(g(x_1, \ldots, x_n, y)=0)$
(III′) 全域的であることが保証される。
　正則関数$g(x_1, \ldots, x_n, y)$から関数$f(x_1, \ldots, x_n)$を以下の操作で作る。
　　$f(x_1, \ldots, x_n)=\mu y(g(x_1, \ldots, x_n, y)=0)$

部分帰納的関数と帰納的関数

部分帰納的(partial recursive)関数
　初期関数(1), (2), (3)に操作(I), (II), (III)を有限回適用して定義された関数。

帰納的(recursive)関数
　(一般帰納的(general recursive)関数)
　初期関数(1), (2), (3)に操作(I), (II), (III′)を有限回適用して定義された関数。

各関数の関係



最後に

- ミニテストを提出してから帰ること
- 次回は、
  - Turing機械